基于改进遗传算法的 SDN 多控制器负载均衡机制研究 *

徐爱鑫 [1,2]，孙士民 [1,2†]，汪晓凡 [1,2]，徐国威 [1,2]，王美玉 [1,2]

(1. 天津工业大学 软件学院，天津 300387; 2. 天津市自主智能技术与系统重点实验室，天津 300387)

摘 要：为解决软件定义网络中多控制器负载失衡问题，提出了一种基于非合作博弈降载的主控制器重选模型。首先，利用动态阈值来判别过载控制器。其次，采用基于优先权的迁移交换机决策机制。最后，构建以控制器集群的负载均衡度、平均总时延和交换机迁移成本作为效用函数的优化模型，采用改进的遗传算法求解，加入相似算子提高寻求全局最优解的速度及准确度。实验结果表明，该机制有效地均衡了控制平面的负载并优化了网络性能。

关键词：软件定义网络；非合作博弈；负载均衡；交换机迁移；遗传算法；算子

中图分类号：TP393 doi: 10.19734/j.issn.1001-3695.2022.02.0070

SDN multi-controller load balancing mechanism based on improved genetic algorithm

Xu Aixin[1,2], Sun Shimin[1,2†], Wang Xiaofan[1,2], Xu Guowei[1,2], Wang Meiyu[1,2]

(1. *School of Software*, *Tiangong University*, *Tianjin* 300387, *China*; 2. *Tianjin Key Laboratory of Autonomous Intelligent Technology & System*, *Tianjin* 300387, *China*)

Abstract: This paper proposed the Non-cooperative Game Theory-based of load-reduction Master Controller Reselection model in Software Defined Networking to address the multi-controller load imbalance problem, called GTMCR. To begin with, the discriminating overload controller utilized the Overload Dynamic Thresholds approach (ODT) . Next, the migration of switches followed a Priority Decision Switching Strategy (PDSS) . Finally, the constructed optimization model took the load balancing degree of controller clusters, average total delay and switch migration cost as utility functions. To prevent the global optimal solution from slipping into the local optimum, the solution value of optimal deployment used the Genetic Algorithm for Improved Multi-objective Optimization (GAIMO) with the addition of a similarity operator to boost the algorithm's convergence speed and the accuracy of the global optimal solution. The experimental results show that this mechanism effectively balances the control plane load while achieving the goal of optimizing network performance.

Key words: software defined networking (SDN); non-cooperative game; load balancing; switch migration; genetic algorithm; operator

0 引言

软件定义网络[1](software defined networking, SDN)是一种新型的互联网体系架构，不仅实现了控制层与数据层的分离，还通过控制器的集中控制来实现对网络灵活便捷的配置和管理。但是，由于网络规模和流量规模的动态变化，控制器可能会发生负载失衡，从而导致网络无法正常运行，过载控制器还可能因此遭受恶意流量的网络攻击，影响网络的正常通信，这给稳定高效的网络服务带来了新的挑战。因此，研究多控制器间的负载均衡具有重要意义。

目前，多控制器部署大多通过静态部署方式来完成[2,3]。在实际通信环境中，网络流量是动态变化的。因此，设计动态的控制器与交换机映射机制，以缩短控制器响应时间并更好地利用控制器资源，对提高 SDN 控制平面的可扩展性和可靠性具有重要作用。刘毅等人[4]针对 SDN 动态流量变化所引起的控制器负载不均衡问题，提出了一种仅以最小迁移代价为目标的阶段式动态负载均衡策略，未考虑其他优化目标。Huang 等人[5]提出了基于聚类的遗传算法与合作集群(CGA-CC)来解决动态控制器部署问题，采用针对所有控制器上的请求分配概率的调度算法来平衡调度性能和可扩展性之间的权衡，没有在指标权重分配上进行细致描述，其算法实现的负载均衡程度有限。Adekoya 等人[6]采用动态映射交换机迁移决策算法 ISMDA(improved switch migration decision algorithm)，通过迁移模型来对迁移交换机的流量容量限制选择最优的控制器部署方案，但需要进一步实验来验证算法的可行性和有效性。

综合考虑目前的控制器部署方案[7]是通过实施多控制器之间的交换机迁移从而实现负载均衡的目标。面临的主要问题为：从控制器的角度来看，如何判定过载状态以及迁移哪台交换机实现主从控制器重映射达到控制平面的负载均衡；从交换机的角度来看，考虑映射到哪台目标控制器来实现负载均衡以及如何设计动态的控制器和交换机的映射目标。

1 相关工作

近年来，对于多控制器中判定过载控制器的研究，Lan 等人[8]提出一种预先定义负载权重的动态自适应的负载均衡策略，设定控制器的权重系数，大幅度减小了通信的流开销，但在控制器响应时间方面表现不佳。Mokhtar 等人[9]提出了多阈值负载均衡切换迁移方案，引入传输负载信息的指示器，通过渐进动态调整阈值来实现控制器之间的持续负载均衡，指示器会增加负载均衡的计算成本且实时性较差。

待迁移交换机的选择大多采用随机原则，传统的方式未

收稿日期：2022-02-18；**修回日期**：2022-04-20 **基金项目**：国家自然科学基金资助项目(61802281，61702366，61972456)；天津市自然科学基金资助项目(19JCYBJC15800)；专用集成电路与系统国家重点实验室(复旦大学)开放课题(2021KF014)

作者简介：徐爱鑫(1997-)，女，河北唐山人，硕士研究生，主要研究方向为软件定义网络、负载均衡；孙士民(1983-)，男(通信作者)，山东临沂人，副教授，博士，主要研究方向为软件定义网络、网络空间安全、QoS 优化算法(sunshimin@tiangong.edu.cn)；汪晓凡(1996-)，女，河北邯郸人，硕士研究生，主要研究方向为区块链、共识算法；徐国威(1998-)，男，江苏宿迁人，硕士研究生，主要研究方向为区块链技术、低延时网络；王美玉(1998-)，女，山西吕梁人，硕士研究生，主要研究方向为区块链技术、低延时网络.

考虑迁入域的负载容易导致二次过载。因此，Xiao 等人[10]提出了一种用于 SDN 控制平面的交换机迁移决策方案(DMSSM),提出了迁移效益模型来解决迁移决策问题,形成迁出域交换机集合，通过负载均衡率和传输时延衡量迁出的交换机，但是未考虑到迁移时的总时延。Sahoo 等人[11]设计了最小化控制器间负载差异为目标的负载均衡算法来选取待迁移交换机，但是未考虑到控制器的负载容量，负载容量小的控制器在迁移后导致过载，因此可能会触发错误的交换机迁移。Zhong 等人[12]提出了一种基于预测的 SDN 负载均衡双重交换机迁移方案。该方案将过去的流量负载作为历史数据来预测未来的流量负载，并减少了交换机迁移的频率，但预测流量需要一定的时间进行预测处理，无疑增加了计算成本。

对于待迁移交换机部署目标控制器的问题，Yuan 等人[13]提出一种基于控制器分配方案的遗传算法(GAPA)来解决控制器的配置问题，该方法在降低 SDN 交换机与控制器之间的平均传播时延方面具有良好的性能，并能实现控制器之间更好的平衡，针对平均传播延迟进行寻最优值从而获得分配结果，但其交叉和变异的遗传算子采用的是传统固定值的方式，从而使得错失最优部署方案，在寻优方面的效果不佳。Mohanty 等人[14]提出了一种有效的基于遗传算法的控制器布局求解方法(Proposed GA)，使传播延迟和优化成本最小化，以找到成本最小的控制器的最优位置，但算法的收敛速度慢，从而导致部署时间太长以及映射策略并不高效。Ibrahim 等人[15]提出了遗传算法(GA)来确定控制器部署的最佳位置和数量，以最小化时延为优化目标，最大限度地减少迭代次数，并确保控制器之间的负载平衡，但算法中的选择算子使得出的策略并不是十分合理。

针对多目标优化选取目标控制器考虑负载均衡方案对网络性能带来的影响，Schütz 等人[16]提出了控制器部署的全面数学形式化，根据传播延迟和控制器的能力确定了交换机与控制器的映射分配关系，同时保持控制器之间的平衡负载分布，但是其只针对平均传播延迟进行寻最优值从而获得分配结果，发送以及处理时延还未考虑进去。Ibrahim 等人[17]提出了一种贪婪随机搜索(GRS)算法用来解决节点与控制器的动态分配问题，通过控制器的位置、数量的分配达到最大的资源利用率以实现负载平衡，使用这样的策略要求控制器的位置正确，以提高网络的可靠性和成本效益，但除了资源利用率外未考虑到其他因素。Zhong 等人[18]提出了评估预测的收益(APOP)方案，这是一种基于过载状态预测和收益评估的多控制器控制平面的负载均衡策略，但它只考虑了负载均衡程度，没有考虑优化通信延迟等其他网络性能因素。

就目前的文献调查情况，为尽可能地减少计算成本和响应延迟，本文对过载控制器选取遵循动态阈值方法。选择待迁移交换机综合考虑了流量变化、负载容量和迁移距离，采用基于优先权的方式进行选取。对目标控制器的选取采用改进的遗传算法，通过引入相似算子加快收敛速度的同时，调整选择、交叉和变异算子确保达到最佳部署策略。结合待迁移交换机与主控制器的重新选取，考虑迁移前后对网络性能的影响，设计的优化目标综合考虑以下指标：控制器上的负载是均匀的，尽可能提升控制器的资源利用率，减少控制平面与数据平面通信的总时延，并尽可能降低交换机迁移的成本。通过以上的交换机迁移方式进行控制平面负载的分配，进而为多控制器负载均衡策略的研究提供解决方案和建模思路。

2　分析与建模

2.1　SDN 分布式网络模型

网络模型 $G=(V,E)$，节点和链路的集合分别为 V、E。网络中的 n 台控制器和 m 台交换机分别表示为 $C=\{C_1,C_2 \ldots C_n\}$ 和 $S=\{S_1,S_2 \ldots S_m\}$。以映射矩阵(1)来表示网络节点之间的链路情况。按照控制器的管控范围划分成若干子域，其子域包括控制器及其管控的交换机。在多控制器架构下，依据交换机访问权限不同，OpenFlow 1.3 协议定义了三种不同角色功能的控制器，它们分别是等价控制器、主控制器和从控制器。其中，主控制器和等效控制器负责管理和控制交换机，但是从控制器只能读取交换机状态而不能对交换机进行管理。

$$\begin{array}{c c} & \begin{array}{cccc} C_1 & C_2 & \cdots & C_n \end{array} \\ \begin{array}{c} S_1 \\ S_2 \\ \vdots \\ S_m \end{array} & \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \end{array} \tag{1}$$

其中，$a_{ij}=\begin{cases} 1, 控制器C_j是交换机S_i的主控制器 \\ 0, 控制器C_j是交换机S_i的从控制器或等价控制器 \end{cases}$

2.2　基于非合作博弈的纯策略纳什均衡模型

非合作博弈是在利益相互影响的局势中如何决策使自己的收益最大，即策略选择问题。对于如何决策目标控制器的部署策略使效用函数值最大的问题，本文对 SDN 网络构建非合作博弈模型。博弈模型可用三元组表示为 $\text{Game}=\{Q,P,U\}$，博弈的三要素如下：

a)参与者 Q：除过载控制器之外，未过载的从控制器作为参与者。式(2)中 C_r 表示过载控制器集合。

$$Q=\{C_t,t\in(1,2\ldots \mathrm{r}-1,r+1\ldots n)\} \tag{2}$$

b)策略集合 P：纯策略是指选择某个策略的概率为 0 或 1，即参与者只使用策略集合中的一条策略。过载控制器 C_r 将控制子域下的交换机 $S_i(i\in(1,m))$迁移至目标控制器 C_k。博弈将在控制器集合 Q 之间进行，在选定博弈对手 C_t 后，每个参与者具有两个纯策略，可用式(3)表示：

$$P_k=\{C_k \text{ accept } S_j, C_k \text{ reject } S_j\} \tag{3}$$

c)效用函数 U：所有参与者在某一组特定策略组合下的效用。它包含控制器的负载、交换机与控制器之间的总时延和交换机的迁移成本。

定义 1　基于控制器负载均衡度的效用函数。在与控制器通信的交互报文中，来自于各交换机的数据报文占主导，如 Packet_In 报文。因此，本文采用控制器处理的 Packet_In 报文数量来计算控制器负载。R_j 为控制器 C_j 单位时间内最大 Packet_In 数据报文处理量，r_i 为交换机 S_i 单位时间内产生的 Packet_In 数据报文的数量，M 为待迁移交换机的个数，则可用式(4)来计算控制器的负载。

$$L_{c_j}=\frac{\sum_{i=1}^{M} r_i\times \mathrm{a}_{ij}}{R_j} \tag{4}$$

控制器的平均负载程度可用式(5)来表示。

$$\overline{L_c}=\frac{\sum_{j=1}^{n} L_{c_i}}{n} \tag{5}$$

为反映 SDN 控制平面的负载均衡分布情况，采用控制器集群负载均衡度 u_1 来衡量，通过标准差得到控制器间的负载差异度，如式(6)。

$$\mathrm{u}_1=\sqrt{\frac{\sum_{j=\infty}^{n}\left(L_{c_1}-\overline{L_c}\right)^2}{n}} \tag{6}$$

定义 2　基于交换机与控制器之间平均时延的效用函数。交换机与控制器之间的时延决定了决策下发的实时性，关系到网络的整体性能。数据报文在交换机与控制器之间的时延主要包括传播时延和网络拥塞时的排队时延。采用交换机到控制器的平均总时延来作为衡量标准如式(7)，其中 T_{ij} 为交

换机 S_i 到控制器 C_j 的总时延。

$$u_2 = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} T_m \times a_i}{m} \tag{7}$$

定义 3 基于交换机迁移平均成本的效用函数。在交换机迁移过程中，实现交换机的控制权转移要经过复杂的角色转换过程。当确定了待迁移的交换机时，控制器在域内将 Flow_Mod 等规则安装到迁移交换机，开销 P_I 如式(8)。

$$P_j = \sum_{i=1}^{m}\sum_{j=1}^{n} a_j \cdot l_j \cdot \tau \tag{8}$$

其中，τ 表示流表安装数据包(包括 Flow_Mod 消息数据包等)的平均大小，l_{ij} 为交换机 S_i 到目标控制器 C_j 的最短可达节点间链路长度。

交换机迁移的通信成本来自于部署流策略产生的代价，它主要受通信需要迁移的消息数目及大小所影响，包括迁移交换机与迁入、迁出控制器之间的通信成本 P_C 如式(9)所示。ε 表示交换机平均通信消息数据包大小(包括 Packet_In 数据包等)，a^0_{ij} 和 a_{ij} 分别表示迁移前后待迁移交换机 S_i 与目标控制器 C_j 之间的映射关系。

$$P_C = \sum_{i=1}^{m}\sum_{j=1}^{n}[(a^0_{ij} \cdot l_j) + (a_{ij} \cdot l_j)] \cdot \varepsilon \tag{9}$$

当迁移交换机向目标控制器发送流请求时，生成迁移请求成本 P_M 如式(10)所示。其中，ζ 为交换机 S_i 进行迁移请求消息数据包(包括 Role-Request 消息数据包等)的平均大小。

$$P_M = \sum_{i=1}^{m}\sum_{j=1}^{n} a_i \cdot l_i \cdot \zeta \tag{10}$$

交换机迁移数可以用式(11)表示为

$$\lambda = \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n}(a^a_a - a_4)^2 \tag{11}$$

本文综合考虑了迁移规则流表安装成本 P_I、通信成本 P_C 和迁移请求成本 P_M。式(12)表示平均迁移成本 u_3。

$$u_3 = (P_S + P_C + P_M)/\lambda \tag{12}$$

若将上述 u_1、u_2、u_3 作为目标函数，则可以将寻找最优控制器-交换机映射矩阵关系问题转换成多目标优化模型。优化模型式(13)是寻找最优解使得 u_i 值最小，即控制器间负载差异、交换机与控制器的总时延和交换机的迁移成本均最小化。

$$\min(u_i), i = 1,2,3 \tag{13}$$

s.t.

$$a_{ij} \in \{0,1\}; \forall i, \forall j, i = 1,2\cdots m, j = 1,2\cdots n$$

$$\sum_{j=1}^{n} a_{ij} = 1; \forall i, i = 1,2,3\cdots, n$$

$$\sum_{i=1}^{m} r_i a_{ij} \le R_j; \forall j, j = 1,2,3\cdots m$$

在该模型中，为了减少集群中交换机频繁交换的额外开支，只要当前控制器还没有出现过载的情况，就不会进行重映射操作。每台控制器在不超负载的情况下尽可能处理更多的数据包，因为这对于控制器来说增大了自身的吞吐量的同时，也减少其余控制器负载的增加，符合纳什均衡博弈中每个参与者的策略是对其他参与人策略的最优反应的原则。因此，该博弈存在纯策略纳什均衡。

对此，本文采用 P_i 表示参与者的某一特定的策略，P_{-i} 代表除参与者以外其余参与者策略的组合，具体表示为 $P_{-i} = (P_1 \cdots P_{i-1}, P_{i+1} \cdots P_n)$。因此，$P = (P_i, P_{-i})$ 表示所有参与者某一特定的策略组合。当博弈趋于稳定时，达到的纳什均衡状态用 P_i^* 表示。对于 n 个参与者的博弈，在其余参与者策略都不变的条件下，对任意一个参与者 i，其策略都是对其余参与者策略组合的最优策略，可以表示为 $P_i^* \in \arg\min_{P_i \in P} u_i(P_i, P_{-i}^*)$，即效用函数值均有不等式 $u_i(P_i^*, P_{-i}^*) \le u_i(P_i, P_{-i}^*)$ 成立。

3 控制器负载均衡模型

针对负载失衡的控制平面，本章首先通过判定过载控制器 ODT 方法，进而利用 PDSS 策略选择过载控制器管控下的交换机进行迁移，迁移至 GAIMO 算法部署目标控制器下，最终通过一个基于非合作博弈降载的主控制器重选模型 GTMCR(game theory-based of load-reduction master controller reselection)对控制器集群中过载控制器进行降载处理，从而使控制平面负载均衡。其算法思想如图 1 所示。

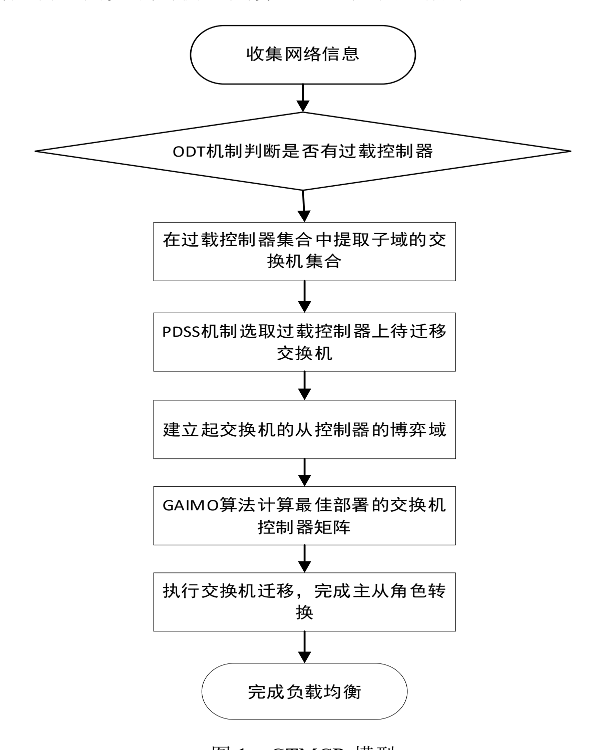


图 1 GTMCR 模型

Fig. 1 GTMCR model

3.1 过载控制器判定策略 ODT

对于过载控制器的判定，本文设计了过载控制器的动态阈值判定机制 ODT (Overload Dynamic Thresholds)，阈值 L_T 根据整体控制平面的负载状况动态调整，定义如式(14)所示。

$$L_T = \begin{cases} \varepsilon & \exists Lc_i \le \varepsilon \\ \frac{1}{n}\sum_{i=1}^{n} Lc_i & \forall Lc_i > \varepsilon \end{cases} \tag{14}$$

其中，ε 为判定过载门限值，Lc_i 为控制器 C_j 的负载情况。M_i 为 0 表示非过载状态，反之处于过载状态，触发 GTMCR 进行负载均衡。控制器状态标志位字段为式(15)。

$$M_i = \begin{cases} 0, L_{cj} \le L_{T_i} \\ 1, L_{cj} > L_{T_i} \end{cases} \tag{15}$$

3.2 迁移交换机选择策略 PDSS

为了尽快地使控制器降载，本文提出一种基于优先权的交换机选择策略 PDSS (Priority Decision Switching Strategy)。若在交换机变更主从控制器关系后有新的大量请求来，则要产生新的流表项向新的目的控制器发送更多的 Packet_In 请求，从而增加业务处理的负担，所以流量变化小的拥有更高的优先迁移权。此外，负载大的交换机进行一次迁移就可完成负载下降，以防迁移的数量过大而增大迁移成本。另外，为了减少待迁移交换机与过载控制器之间的通信成本，距离目标控制器越近被迁移的优先权也越高。因此，优先迁移流量变化小、负载占比较高且与目标控制器之间距离较近的交换机。

式(16)表示 S_i 在 C_j 上所占用的控制资源 X_{ij}。其中，S_i 加

入 C_j 的事件数用 $\lambda(S_i)$表示，$B_{ij}(s_i)$代表 S_i 的主控制器 C_j 的总事件数。

$$\chi_{ij}=\frac{\lambda(s_i)}{B_{ij}(s_i)} \tag{16}$$

因此，式(17)定义 S_i 迁移到 C_r 的优先权为 Pr_{ir}。它们之间的跳转数目用 d_{ir} 表示，S_i 的流变化速率为 V_i。

$$\mathrm{Pr_{ir}}=\frac{\dfrac{\chi_{ir}}{d_{ir}V_i}}{\sum\limits_{s_k\in s_{c_r}}\dfrac{\chi_{kr}}{d_{kr}V_k}} \tag{17}$$

式(18)为选取的迁移交换机 S_i。

$$S_i=\max\left\{\mathrm{Pr}_{s_j},S_j\in S\right\} \tag{18}$$

3.3　基于改进遗传算法的控制器决策机制 GAIMO

遗传算法[19](Genetic Algorithm，GA)是一种启发式的搜索算法，通常用来解决优化问题。它依靠概率性优化搜索，能够自适应地修正和指导优化的搜索空间。本节提出面向改进多目标优化的遗传算法 GAIMO (Genetic Algorithm for Improved Multi-objective Optimization)，利用遗传算法对网络负载均衡问题进行全局搜索，然后在全局最优附近区域进一步局部寻优，最终找到网络负载均衡最优解。

3.3.1 GAIMO 算法

GAIMO 算法在基础的遗传算法上加入了相似算子的概念，并采用两种方式相结合的选择算子以及自调节的交叉和变异概率。算法的过程包括：个体编码、种群初始化、个体评价、选择、相似度判断、交叉、变异操作以及种群迭代更替，具体设定如下：

整个网络里的交换机和控制器用 $m\times n$ 维矩阵表示。采用随机序列初始化使产生的个体具有一定的差异度，保证均匀性的同时提高种群的多样性。个体适应度函数用于区分群体中个体好坏，适应度值越大表示控制器的性能越能得到充分的发挥。针对多目标优化问题采用离差标准化方法 (Min-max Normalization)，通过转换函数对原始数据进行线性变换。适应度函数如式(19)所示。其中 u_{imax} 和 u_{imin} 为个体的最大值和最小值。对于多目标优化适应度函数式(20)，采用的权重因子为 w_1 、w_2 、w_3。

$$f_i(x)=\frac{u_{i\max}-u_i}{u_{i\max}-u_{i\min}},u_{i\min}\le u_i\le u_{i\max},i=1,2,3 \tag{19}$$

$$F(x)=w_1\cdot f_1(x)+w_2\cdot f_2(x)+w_3\cdot f_3(x) \tag{20}$$

$$\text{s.t.}\quad w_1+w_2+w_3=1$$

选择算子是为了在种群中选择创造下一代的个体。本文的选择算子采用 5%精英策略，先选出最佳个体遗传到下一代，为了保证亲本的多样性，再结合轮盘赌方法。式(21)表示个体 x_i 被选择的概率。

$$P(x_i)=\frac{f(x_i)}{\sum\limits_{i=1}^{N}f(x_i)} \tag{21}$$

由于近亲交叉繁殖会增加不必要的迭代计算[20]，在传统的遗传算法基础之上，引入了相似算子，用海明距离 GH_{ij} 衡量两个即将进行交叉个体的相似程度，海明距离是指相同长度的以 a 为基的两个字串 i 和 j 中对应位不相同的数量。因此，先对交叉个体进行相似度评定，低于相似度阈值时，再给予交叉机会。这样可以尽可能地减少迭代次数，实现保证解的准确度的同时从而加快收敛速度。

交叉为了让后代能够继承上一代的优良基因，在固定概率 P_c' 的基础上设定与进化代数相关的交叉概率，通过交叉概率来加快种群的收敛以便加速寻找优良种群所处的区域。由于 P_c 的取值过大容易使适应问题环境值高的基因串很快被破坏掉，所以 P_c 值随遗传代数的增加而逐渐递减，这样同时也解决了取值过小而使搜索速度缓慢的问题。增加交叉点会降低性能。因此，为保持多样性，本文采用两点交叉算子(式(22))。其中，Gen 为最大迭代次数，当前迭代次数为 i。

$$p_c=\begin{cases}p_c'\cdot\left(1-\dfrac{i}{\mathrm{Gen}}\right) & i\ge \mathrm{Gen}\\ p_c' & i<\mathrm{Gen}\end{cases} \tag{22}$$

为保持种群遗传多样性，避免落入局部极大值陷阱，采用择位变异和自适应变异概率 P_m(式(23))。这种方法的优势在于增加劣势个体变异概率的同时，减小优秀个体的变异概率。因此，减小原有变异概率 P_m'，在优良种群周围进行小步距大概率变异。这样既避免了算法陷入局部极小又极大地降低再次产生无用的劣质解发生的概率，加快收敛速度。随着进化过程的推进，概率逐渐减小趋于稳定。这避免了对算法后期的稳定性造成冲击而导致算法收敛时间变长，提高了计算的效率。其中，f'、f_{max} 和 f_{avg} 分别为当前、最大和平均适应度值。

$$p_m=\begin{cases}p_m'\cdot\left(1-\dfrac{f_{\max}-f'}{f_{\max}-f_{arg}}\right) & f\ge f_{a_g}\\ p_m' & f<f_{a_g}\end{cases} \tag{23}$$

经过不断地种群更替，最终算法收敛到全局最优解。算法的程序流程图如图 2 所示。

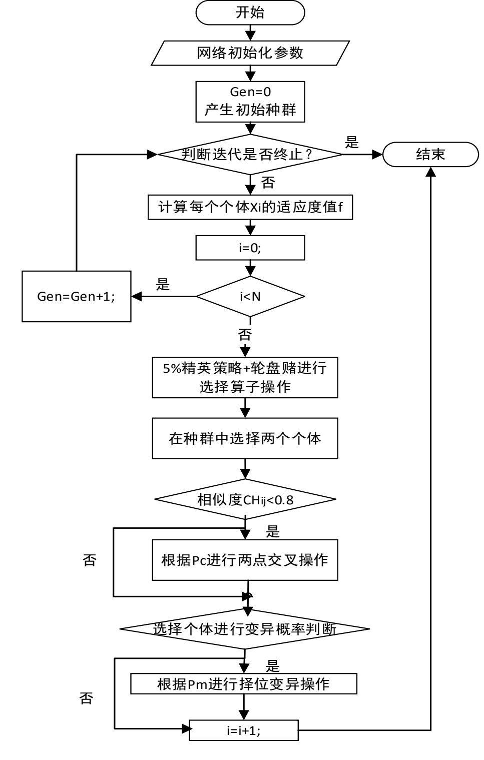


图 2　GAIMO 算法流程图

Fig. 2　GAIMO algorithm flow chart

3.3.2 GAIMO 算法的实现

本文提出的 GAIMO 算法是基于遗传算法对交换机与控制器的映射关系进行寻优。首先确定算法执行所需要的输入参数，然后对种群进行二进制编码设计，对每一个个体进行初始化，并对初始种群中的每一个个体计算适应度函数值(第 a~b 行)。接下来开始进行遗传算法的算子操作(第 c~s 行)，先进行选择算子操作,采用精英策略与轮盘赌选择相结合的方式(第 e~j 行)。本文提出加入相似算子进行相似度条件判断，若相似度低于相似度阈值，则进行自适应两点交叉操作(第 k~n 行)。接下来进行自适应变异概率的遗传操作(第 o~p 行)。形成新的种群(第 q 行)之后，对上述算子不断地迭代进行种群更替，最后得到的最优个体就是对遍历到待迁移交换机 S_i 重选的主控制器(第 r 行)。算法描述如下：

算法 1　GAIMO 算法

输入：网络参数，最大迭代次数 *Gen*，种群大小 *N*，交叉概率 *crossover_p* 和变异概率 *mutate_p*;

输出：负载均衡度、交换机与控制器之间的平均时延和交换机的迁移成本开销；最优个体的适应度函数值，最优个体的交换机与控制器的映射矩阵。

a) $pop = initialpop(N, chromlength, len, m)$;//初始化种群

b) $Functionvalue = objectiveFunction(Population)$;/*适应度函数值计算*/

c) for $i = 1:Gen$

d) $fitvalue = cal_objective(pop, r, R, popsize, len, m)$;

e) $Newpop = selection(pop, fitvalue)$;//选择算子

f) if $i <= 0.05*N$

g) $Elitist(X_i) = Max(pop)$;//精英策略

h) else

i) $P(X_i) = f(X_i)/sum(f)$;//比例选择策略

j) end if

k) if $GH_{ij} < 0.8$ //相似度判断

l) $p_c = adapt_cross(crossover_p)$;//自适应交叉概率

m) $newpop = crossover(newpop, p_c, len, m)$;//交叉

n) end if

o) $p_m = adapt_mutate(mutate_p)$;//自适应变异概率

p) $newpop = mutation(newpop, p_m, len, m)$;//变异

q) $pop = newpop$;//种群更替产生新种群

r) $[bestindividual, bestfit] = best(pop, fitvalue)$;//择优

s) end for

3.4　基于非合作博弈降载的主控制器重选模型 GTMCR

在 GTMCR 算法的执行过程中，首先遍历所有控制器，对各个控制器的负载进行监测，通过动态负载阈值判定过载控制器，决策出需要迁移交换机的控制器集合，进而建立目标控制器的博弈域(第 a~h 行)。然后依据交换机的迁移优先权，选择过载控制器管控下优先权最大的交换机进行迁移(第 l~n 行)。接下来重选待迁移交换机的主控制器，在控制器间的非合作博弈中，寻求目标控制器的纯策略纳什均衡状态，通过上述的 GAIMO 算法得到所要迁移到的目标控制器，将迁移交换机的主从控制器进行角色转换，实现过载控制器降载(第 o~r 行)。最终达到新的负载均衡，得到迁移后控制平面的负载均衡度(第 u 行)。最终根据迁移情况，构建新的 SDN 网络拓扑。算法描述如下：

算法 2　GTMCR 算法

输入：网络状况参数，如交换机数量 *m* 与控制器数量 *n*、控制器 C_j 的处理速率 R_j、交换机 S_i 的请求速率 r_i、控制器与交换机之间的时延 *T* 和距离 *D* 等相关参数。

输出：网络中过载控制器，控制器集群的负载均衡度，网络中需要迁移的交换机。

a) for each C_j

b) $Lc_i = LoadFunction(c_j)$;//计算控制器负载

c) if $Lc_j < LT$ return 0;/*决策是否为过载控制器,是否进行交换机迁移*/

d) else

e) M_i=1； //过载控制器标志位

f) $C = ODT(Lc_j)$;//决策过载控制器集合 *C*

g) end if

h) end for

i) for C_i in *C* //遍历过载控制器

j) $S = traversal(C_i, L_{ij}, X_{ij})$ ； //控制器管控的交换机集合

k) for S_i in *S*

l) $Pr_{ir} = Pr()$;//计算 S_i 迁移到 C_r 的优先权

m) $S_i = \max\{\Pr_{s_j}, s_j \in S\}$;//优先权最大进行迁移

n) $Migrate(s_i) = PDSS(S)$;//决策迁移交换机

o) $U = U_function(C_i)$ ； //计算效用函数

p) $Game = \{C, P, U\}$;//建立博弈模型

q) $C_r = GAIMO(S_i)$ ； //决策目标控制器

r) $Master(S_i) = C_r$ ； //主从控制器角色转换

s) end for

t) end for

u) $Lc = sum(L_{Cj})/n$ ； //迁移后负载均衡度的计算

4　仿真实验

4.1　实验环境设定

实验环境配置为 Intel i5 处理器，内存 4GB，CPU 3.40GHz，Windows10 操作系统，使用仿真工具 MATLAB R2017b，控制器容量设定为 10MB，交换机与控制器间时延和链路距离依据实验拓扑得出。实验仿真参数值参照流量特征数值[21]，控制器单位时间的处理 Packet_In 流请求数据包为 10000 个左右。实验采用 the Internet topology zoo[22] 中具有一定规模差异的网络拓扑，具体网络信息如表 1 所示。

表 1 不同网络规模的节点链路信息

Tab. 1　Node link information for different network sizes

网络	OS3E	Ntelos	IRIS	Interllifiber	Interoute
节点数	34	48	51	73	110
链路数	42	61	64	93	149
控制器数	5	6	6	7	10

4.2　实验与结果分析

为了验证本文提出的 GAIMO 算法的有效性与可行性，本节将 GAIMO 算法与现有解决控制器负载均衡问题的 GAPA 算法[13]、Proposed GA 算法[14] 以及 GA 算法[15]在网络负载均衡度、平均总时延、迁移成本的性能上进行对比。设置固定交叉概率为 0.4，变异概率为 0.04，迭代次数为 200。对于适应度函数值进行归一化，同时为了突出重要因素对于控制平面与数据平面的的影响，选取的权值比重 w_1、w_2、w_3 分别为 0.5、0.3 和 0.2。

4.2.1 控制器负载均衡度对比

在实验开始后，每个交换机发送数据报文到 OS3E 网络部署的 5 台控制器，整个网络被划分为五个子域。等待 10 秒后，所有控制器获得状态初始值。在此时，控制器的负载较为均衡，负载比为 2:4:3:2:3(图 3)。随后，增加交换机和控制间的交互报文量，使控制器负载逐步上升。从图中可以看出控制器 C_2 和 C_4 超出阈值线，先后出现了过载现象，在第 40 秒时负载比达到了 10:13:11:14:8，通过式(6)计算得到此时负载均衡度为 2.1354。因而，GTMCR 机制触发，进行交换机迁移。之后，控制器之间负载差明显收敛。交换机迁移结果是：C_2 下的部分交换机迁移到 C_1，C_4 下的部分交换机迁移到 C_5，迁移之后的负载比为 23:23:22:22:22，负载均衡度为 0.4899，使得控制平面负载达到了新的平衡。

在图 4 中，本文对 GA、GAPA、Proposed GA 和 GAIMO 算法在控制器负载均衡适应度上的表现进行对比。纵坐标表示控制器负载均衡适应度函数值，反映控制器集群的负载均衡的情况。适应度数值越高，表示控制器负载均衡性能越好。为验证四种算法对控制器集群负载的均衡效果，采用了基于 OS3E 的网络拓扑，上述四种算法的适应度函数都有所提升，说明在负载失衡的情况下均能够通过交换机迁移的方式在一定程度上实现负载均衡。GA 和 GAPA 在 200 代的进化过程中容易出现大的波动，收敛性差，原因是交叉和变异概率固定不变难以收敛。Proposed GA 的负载均衡度的适应度值从 0.813 提升到 0.915。相对而言，GAIMO 收敛最快，负载均

衡度的适应度值从 0.813 提升到 0.95。可见，GAIMO 算法能够更有效地均衡控制平面负载。

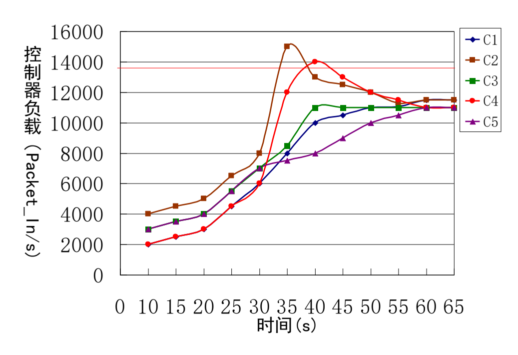


图 3　迁移前后控制器负载变化情况

Fig. 3　Change in controller load before and after migration

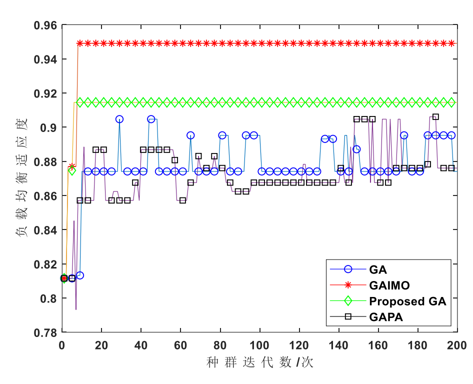


图 4　控制器负载均衡对比

Fig. 4　Comparison of average switch-to-controller latency

为测试拓扑大小对算法性能的影响，本文采用的测试环境为 5 种不同网络规模的拓扑结构。图 5 中纵坐标为负载均衡度，值越小表明控制平面负载越均衡。横坐标为不同的网络拓扑。在较小的 OS3E 网络下，四种算法的负载均衡度差异较大。这是由于在较小的拓扑下，交换机节点较少，控制器间会因为管理交换机的数量不同而导致负载差异大，进而使负载均衡参数值偏大。随着网络规模的增大，GA 和 GAPA 算法负载均衡度相近，均衡效果并不佳。相对而言，GAIMO 算法随拓扑的增大负载均衡效果更明显，得到的负载均衡度始终比其他算法小，且变化趋势较为平缓。由于 GA、GAPA 和 Proposed GA 容易收敛到局部最优解，而 GAIMO 算法收敛到全局最优解，使得负载分配更加合理，网络负载均衡度参数也越来越小。实验结果也证明了 GA、GAPA 和 Proposed GA 并不适用于小型网络中，而 GAIMO 能够在各种规模的网络上保持较好的负载均衡性能。

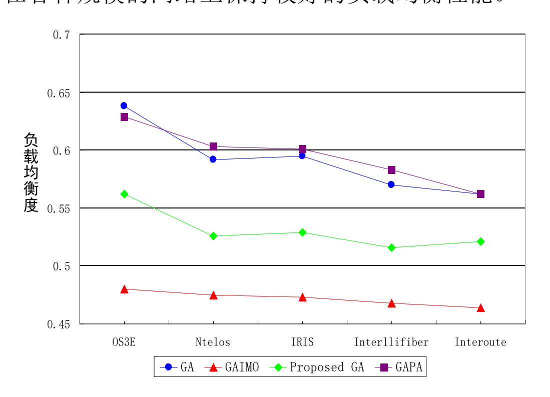


图 5　不同网络规模的控制器负载均衡度

Fig. 5　Controller load balancing degree for different network sizes

4.2.2 交换机到控制器的平均总时延

为对比不同算法下交换机到控制器之间通信总时延的优化效果，实验采用 OS3E 网络测试时延适应度函数的优化效果。从图 6 可以看出，Proposed GA、GAPA 和 GA 算法中的最高适应度函数值最终只能收敛到 0.85，均低于 GAIMO 算法，说明找到的映射方案并不是最佳的。GAIMO 算法不仅收敛速度最快而且找到了最优解，更好地减少了迁移的总时延。在平均时延的控制上，GAIMO 算法均优于其他算法且优化效果更加稳定，能够较大幅度地缩短交换机到控制器之间的平均时延，更有效地减小通信开销，保证网络通信质量。

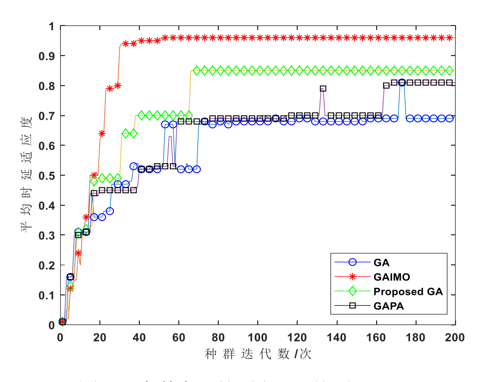


图 6　交换机到控制器平均时延对比

Fig. 6　Average Switch to Controller Latency Comparison

4.2.3 交换机的迁移成本

在交换机迁移过程中会产生大量的通信消息数据包，为实现控制器平面负载均衡的同时尽可能地减少迁移成本，实验采用 OS3E 网络测试不同算法下迁移成本的大小。测试结果如图 7 所示。GA、Proposed GA 算法由于选择策略过早收敛从而错失了最优解，而 GAPA 算法一直都未收敛，原因是交叉和变异算子是固定值，无法根据迭代状况作出相应的调整而难以收敛。GAIMO 算法收敛速度更快，更容易找到最优解，目标函数值最终达到了 0.9，是因为改进的算子结合迭代次数和适应度函数值的变化动态调整，达到更高效地找到全局最优解的目的。GAIMO 算法的迁移策略效果最佳，能够大幅度减少网络中多台交换机迁移的成本代价，证明了用交换机迁移方法提升网络性能的有效性。

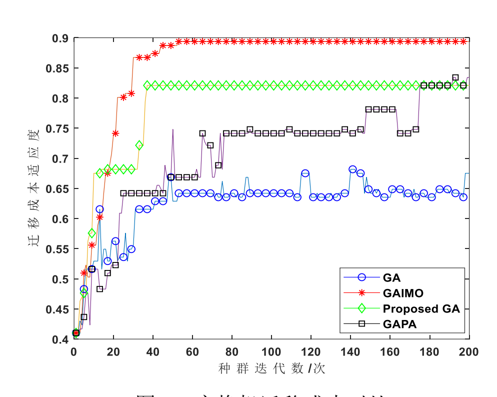


图 7　交换机迁移成本对比

Fig. 7　Switch migration cost comparison

4.2.4 多目标性能优化

多目标优化适应度函数值反映控制器负载均衡的情况，适应度数值越高，控制器负载均衡的网络性能越好。如图 8 所示， GAIMO 算法从 20 代开始，最优解的适应度值相比于其他算法更优且收敛速度最快。这是由于 GAIMO 算法采用相似算子，尽量避免下一代中出现近亲繁殖，从而减少重复交叉和交叉无效的情况，减少了时间复杂度。GAIMO 算法从 50 代开始稳定且找到最优部署。这是因为自适应的交叉概率考虑迭代次数以及变异概率考虑适应度函数值的偏好情况，突变的发生随着遗传代数和适应度函数值的增大而随之减少。GAIMO 算法最终的收敛值最大，说明 GAIMO 采用的精英策略防止了最优解丢失从而保证种群向优进化的趋

势并趋于稳定达到 0.94。因此，GAIMO 算法更能对激增的流量负载进行均衡的同时，提升控制平面的鲁棒性。

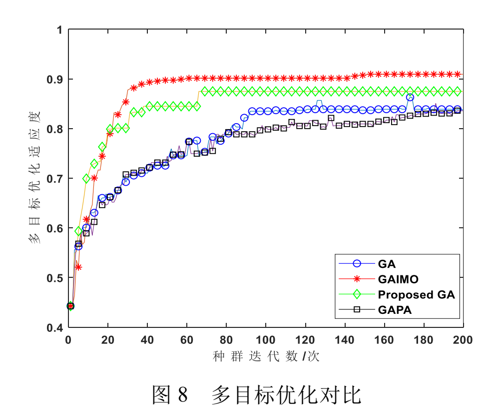


图 8　多目标优化对比

Fig. 8　Multi-objective optimization comparison

5　结束语

本文构建了多目标优化模型 GTMCR，对于判别过载控制器采用动态阈值 ODT 算法，迁移交换机的决策采用基于优先权的迁移策略 PDSS 算法，同时设置了相应的约束条件，采用改进的遗传算法 GAIMO 获得交换机迁移的最佳策略。实验结果表明，GAIMO 算法更加有效地提升收敛速度，能最快找到最优部署，合理地平衡控制器集群负载均衡度，提升了全局网络性能。接下来，需要将重映射过程中控制器与交换机以及交换机间通信会出现丢包等容错情况也考虑进来，如果当控制器负载严重过载或出现故障时，除了依靠交换机迁移，还需添加额外的控制器，进一步研究不同情况下的负载均衡。

参考文献：

[1] Isong B, Molose R R S, Abu-mahfouz A M, *et al.* Comprehensive Review of SDN Controller Placement Strategies [J]. IEEE Access, 2020, 8: 170070–170092.

[2] Hamdan M, Hassan E, Abdelaziz A, *et al.* A comprehensive survey of load balancing techniques in software-defined network [J]. Journal of Network and Computer Applications, 2021, 174: 102856.

[3] Alowa A, Fevens T, Khamayseh Y. Survival backup strategy for controller placement problem in Software Defined Networking [J]. Computer Communications, 2022, 185: 104-117.

[4] 刘毅，李凯心，李国燕，等. 基于 SDN 的动态负载均衡策略 [J]. 计算机应用研究, 2020, 37 (10): 3147-3152. (Liu Yi, Li Kaixin, Li Guoyan, *et al.* Dynamic load balancing strategy based on SDN [J]. Application Research of Computers, 2020, 37 (10): 3147-3152.)

[5] Huang V, Chen G, Zhang P, *et al.* A Scalable Approach to SDN Control Plane Management: High Utilization Comes With Low Latency [J]. IEEE Trans on Network and Service Management, 2020, 17 (2): 682-695.

[6] Adekoya O, Aneiba A, Patwary M. An Improved Switch Migration Decision Algorithm for SDN Load Balancing [J]. IEEE Open Journal of the Communications Society, 2020, 1: 1602–1613.

[7] Shirmarz A, Ghaffari A. Taxonomy of controller placement problem (CPP) optimization in Software Defined Network (SDN): a survey [J]. Journal of Ambient Intelligence and Humanized Computing, 2021, 12 (12): 10473–10498.

[8] Lan W, Li F, Liu X, *et al.* A Dynamic Load Balancing Mechanism for Distributed Controllers in Software-Defined Networking [C]// the 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) . Changsha: IEEE, 2018: 259–262.

[9] Mokhtar H, Di X, Zhou Y, *et al.* Multiple-level threshold load balancing in distributed SDN controllers [J]. Computer Networks, 2021, 198: 108369.

[10] Xiao H, Hu B, Zhou L, *et al.* DMSSM: A Decision-Making Scheme of Switch Migration for SDN Control Plane [C]// IEEE the 7th International Conference on Computer Science and Network Technology (ICCSNT) . Dalian, China: IEEE, 2019: 348-352.

[11] Sahoo K S, Puthal D, Tiwary M, *et al.* ESMLB: Efficient Switch Migration-Based Load Balancing for Multicontroller SDN in IoT [J]. IEEE Internet of Things Journal, 2020, 7 (7): 5852-5860.

[12] Zhong H, Xu J, Cui J, *et al.* Prediction-based dual-weight switch migration scheme for SDN load balancing [J]. Computer Networks, 2022, 205: 108749.

[13] Yuan T, Huang X, Ma M, *et al.* Balance-Based SDN Controller Placement and Assignment with Minimum Weight Matching [C]// IEEE International Conference on Communications (ICC) . Kansas City, MO: IEEE, 2018: 1–6.

[14] Mohanty S, Priyadarshini P, Sahoo S, *et al.* Metaheuristic Techniques for Controller Placement in Software-Defined Networks [C]// TENCON-IEEE Region 10 Conference (TENCON) . Kochi, India: IEEE, 2019: 897–902.

[15] Ibrahim A A Z, Hashim F, Sali A, *et al.* A Modified Genetic Algorithm for Controller Placement Problem in SDN Distributed Network [C]// the 26th IEEE Asia-Pacific Conference on Communications (APCC) . 2021: 83-88.

[16] Schütz G, Martins J A. A comprehensive approach for optimizing controller placement in Software-Defined Networks [J]. Computer Communications, 2020, 159: 198-205.

[17] Ibrahim A A Z, Hashim F, Noordin N K, *et al.* Heuristic Resource Allocation Algorithm for Controller Placement in Multi-Control 5G Based on SDN/NFV Architecture [J]. IEEE Access, 2021, 9: 2602-2617.

[18] Zhong H, Fan J, Cui J, *et al.* Assessing Profit of Prediction for SDN controllers load balancing [J]. Computer Networks, 2021, 191: 107991.

[19] Katoch S, Chauhan S S, Kumar V. A review on genetic algorithm: past, present, and future [J]. Multimedia Tools and Applications, 2021, 80 (5): 8091–8126.

[20] Bao L, Zhang L, Sun T. Research on assembly line scheduling based on small population adaptive genetic algorithm [C]// the 6th International Conference on Intelligent Computing and Signal Processing (ICSP) . 2021: 166-170.

[21] Isyaku B, Mohd Zahid M S, Bte Kamat M, *et al.* Software Defined Networking Flow Table Management of OpenFlow Switches Performance and Security Challenges: A Survey [J]. Future Internet, 2020, 12 (9): 147.

[22] Knight S, Nguyen H X, Falkner N, *et al.* The Internet Topology Zoo [J]. IEEE Journal on Selected Areas in Communications, 2011, 29 (9): 1765-1775.

[23] 徐爱鑫　天津市西青区宾水西道 399 号天津工业大学 300387 13230847811 13231568733@163. com

[24] 孙士民　天津市西青区宾水西道 399 号天津工业大学 300387 18722128913 sunshimin@tiangong. edu. cn

[25] 汪晓凡　天津市西青区宾水西道 399 号天津工业大学 300387 15122708693 wxf241382009@163. com

[26] 徐国威　天津市西青区宾水西道 399 号天津工业大学 300387 17519301800 1921994090@qq. om

[27] 王美玉　天津市西青区宾水西道 399 号天津工业大学 300387 17835204887 1714116205@qq. com